Applicant: Beecroft et al.
Application No.: 10/714,696

REMARKS/ARGUMENTS

Claims 1 – 21 are currently pending in this application.

Claim Rejections - 35 USC §102

Claims 1 – 21 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 7,133,940 (Blightman).

Blightman teaches a network interface device which employs a DMA command queue. To avoid the interface processor monitoring multiple DMA commands (to ensure that they are executed), a command queue is maintained by the interface device. The queue contains DMA commands or address values/pointers (identifying where a DMA command is being stored). The DMA commands in the queue are pushed and popped to ensure fast turn around.

In the case of multiple DMA command, with Blightman the network interface processor does not monitor the execution of each DMA command. Instead, it only looks out for the value/pointer for the last DMA command of a multiple group of commands after completion. Also, although Blightman does mention the possibility of using more than one DMA command queue with separate command controllers for each queue, Blightman is silent as to the purpose or manner of operation of such multiple queues. More importantly, there is no suggestion in Blightman of assigning memory space in the DMA command queue to a specific user process.

Instead, with Blightman, the DMA commands will be pushed and popped into and out of the command queue in the order that they are received, irrespective of the user process they relate to.

Furthermore, Blightman does not describe in detail how an individual DMA command is executed – Blightman is only concerned with the concept of storing DMA commands at the network interface. It must be assumed, therefore, that the manner in which the DMA commands are executed is conventional. That is to say, in order to maintain the security of multiple programs (user processes), i.e., to ensure that data for one user process is not mixed up with data for a different user process, each DMA command can only be executed by the CPU and it is the CPU which maintains the security of the individual processes.

The present invention arises from the realization that latency in a large scale parallel processing system can be considerably reduced by changing the manner in which DMA commands are executed. Thus, with the present invention, as claimed in claim 1, the DMA commands are stored in a queue at the network interface. However, unlike Blightman, areas of the network interface command queue are allocated *exclusively* to a program/user process. This means that only those DMA command pertaining to a specific user process can be written to that memory area of the command queue and each user process is allocated its own memory area. The result of this is that the command queue for each user process is managed

independently of any other command queues. This also results in the removal of the need for the DMA commands to be executed by the CPU. Instead, the exclusive allocation of a memory area for the DMA commands of a specific user process ensures the security of multiple user processes running simultaneously on the same CPU.

Respectfully submitted,

Beecroft et al.

By _____
Darryl W. Shorter
Registration No. 47,942

Volpe and Koenig, P.C.
United Plaza, Suite 1600
30 South 17th Street
Philadelphia, PA 19103
Telephone: (215) 568-6400
Facsimile: (215) 568-6499

DWS/rlm
Enclosure